AMENDMENTS

AMENDMENTS TO THE CLAIMS

1-51. (Cancelled)

52. (Currently Amended) A method ~~for debugging a program in real time and while said program is executed by a programmable logic controller, said method~~ comprising ~~the steps of~~:

regarding an entire program stored in a first section of memory and executed by a programmable logic controller, while said entire program is executing and without significantly interfering with execution timing of said program:

displaying a section of said program indicated by a user to be debugged, said section comprising fewer instructions than said entire program;

~~saving original compiled code of said program;~~

compiling said section of said program to be debugged in ~~another~~ a second section of memory;

jumping to said another section of said memory during execution of said program when an instruction indicated to be debugged is to be executed; and

capturing a status of said instruction as it is executed~~, wherein said program is debugged in real time and while said program is executed by~~ said ~~programmable logic controller~~.

53. (Original) The method according to claim 52, further comprising the step of returning to said original compiled code of said program after said instruction indicated to be debugged is executed.

54. (Previously Presented) The method according to claim 53, further comprising the step of restoring said original compiled code once said status is captured.

55. (Original) The method according to claim 52, further comprising the step of

instrumenting each instruction compiled in said another section of memory.

56. (Previously Presented) The method according to claim 52, further comprising the step of storing a table relating instructions to boolean expressions, wherein said instructions are debugged with said boolean expressions.

57. (Previously Presented) The method according to claim 52, further comprising the step of providing a table of pointers to instructions of said original compiled code, wherein said instructions are located in memory during debugging.

58. (Previously Presented) The method according to claim 52, further comprising the step of limiting a data size of each compiled instruction, wherein execution of said instructions to be debugged is faster and memory required to store said instructions is reduced.

59-83. (Cancelled)

84. (New) The method of claim 52, further comprising:

providing a table relating instructions to boolean expressions, wherein said section of said program is debugged utilizing said boolean expressions.

85. (New) The method of claim 52, further comprising:

providing a table of pointers to instructions of said entire program.

86. (New) The method of claim 52, further comprising:

providing a machine code instruction adapted to save a power flow status associated with said section of said program.

87. (New) The method of claim 52, further comprising:

providing a machine code instruction adapted to save an operand value associated

with said section of said program.

88. (New) The method of claim 52, further comprising:

comparing a scan count status word to a current value of a scan counter to determine that said status came from a single scan cycle.

89. (New) The method of claim 52, further comprising:

copying a scan counter value to a scan count status word to determine that said status came from a single scan cycle.

90. (New) The method of claim 52, further comprising:

comparing a scan count status word to a current value of a scan counter to determine that said status came from a single scan cycle

clearing a flag in a buffer if said scan count status word is different from said current value of said scan counter.

91. (New) The method of claim 52, further comprising:

acquiring results from an execution of said section of said program.

92. (New) The method of claim 52, further comprising:

executing said section of said program.

93. (New) The method of claim 52, further comprising:

displaying results from an execution of said section of said program on a human machine interface of said programmable logic controller.

94. (New) The method of claim 52, further comprising:

determining a status window size by a number of operand values returned from an execution of said section of said program.

95. (New) A machine-readable medium having stored thereon a plurality of executable instructions, the plurality of instructions comprising instructions for:

regarding an entire program stored in a first section of memory and executed by a programmable logic controller, while said entire program is executing and without significantly interfering with execution timing of said program:

displaying a section of said program indicated by a user to be debugged, said section comprising fewer instructions than said entire program;

compiling said section of said program to be debugged in a second section of memory;

jumping to said another section of said memory during execution of said program when an instruction indicated to be debugged is to be executed; and

capturing a status of said instruction as it is executed.

96. (New) A circuit embodying a plurality of executable instructions, the plurality of instructions comprising instructions for:

regarding an entire program stored in a first section of memory and executed by a programmable logic controller, while said entire program is executing and without significantly interfering with execution timing of said program:

displaying a section of said program indicated by a user to be debugged, said section comprising fewer instructions than said entire program;

compiling said section of said program to be debugged in a second section of memory;

jumping to said another section of said memory during execution of said program when an instruction indicated to be debugged is to be executed; and

capturing a status of said instruction as it is executed.